

AMENDMENTS

In the claims:

1. (currently amended) A method of switching an Ethernet packet, the method comprising:

computing a plurality of tags for the Ethernet packet, each of said plurality of tags computed using two or more fields in said packet, wherein said fields are selected from Ethernet and network headers in said packet and said plurality of tags corresponds to a plurality of flow detectors in a flow switch node;

looking up the computed plurality of tags in a table, the table containing entries associated with tags, the entries associating switching information with a tag, said switching information defining a route through a plurality of interconnected switch nodes; and

using said switching information associated with one of the computed plurality of tags from the table to switch the packet, if there is an entry for at least one of the computed plurality of tags in the table.

2. (previously presented) The method of claim 1, wherein the method of switching an Ethernet packet further comprises sending the packet to a system with resources for routing a packet and for determining switching information.

3. (previously presented) The method of claim 2, further comprising updating the table to include an entry for the computed tag, and wherein the computed tag is associated with the determined switching information.

4. (previously presented) The method of claim 2, further comprising including an entry in the table for the computed tag associated with a switching instruction indicating that packets should be dropped until the determining of switching information is complete.

5. (previously presented) The method of claim 1, wherein the entries in the table are removed if the tag corresponding to the entry has not been looked up in a predetermined period.

6. (original) The method of claim 5, wherein the length of the tag is determined by the predetermined period, the number of entries in the table, and the probability of two packets generating the same tag.

7. (canceled).

8. (previously presented) The method of claim 1, wherein a plurality of tables are maintained, each table corresponding to one of the flow detectors.

9. (previously presented) The method of claim 1, wherein each of the tags in the plurality of tags includes information about the associated flow detector.

10. (previously presented) The method of claim 1, wherein an error rate of the method is measured based on the number of matches between tags in the table without regard to which flow detector is associated with a tag.

11. (original) The method of claim 10, wherein a warning is issued when the error rate exceeds a predetermined level.

12. (previously presented) The method of claim 5, wherein the predetermined period for which entries in the table are retained without being looked up is decreased when the error rate increases above a predetermined level.

13. (previously presented) The method of claim 5, wherein the predetermined period for which entries in the table are retained without being looked up is increased when the error rate decreases below a predetermined level.

14. (previously presented) The method of claim 1, wherein the plurality of tags are computed in parallel by the plurality of flow detectors.

15. (previously presented) The method of claim 1, wherein each of the plurality of tags computed by the plurality of flow detectors are the same length.

16. (previously presented) The method of claim 1, wherein the plurality flow detectors are associated with a priority, and wherein the switching occurs according to the priority of the flow detector.

17. (previously presented) The method of claim 1, wherein the error rate of the switching system is measured based on the number of cross flow detector tag matches in the table.

18. (original) The method of claim 1, wherein the computing further comprises using a mask of bits of the packet as a seed for a hash code generator.

19. (original) The method of claim 18, wherein the hash code generator is a pseudo random number generator.

20. (original) The method of claim 18, wherein the hash code generator is a shift register with a feedback loop.

21. (original) The method of claim 18, wherein the hash code generator has a non-zero probability of generating the same tag from different input packets.

22. (original) The method of claim 18, wherein the length of the tag is determined by the probability of the hash code generator producing the same hash code from different input packets.

23. (previously presented) A method comprising:

 computing a tag for an Ethernet packet, said tag computed using at least two fields in said packet, wherein said fields are selected from Ethernet and network headers in said

packet_and a length of said tag is determined by the probability of a hash code generator producing an identical hash code from different input packets;

looking up the computed tag in a table, the table comprised of entries, the entries associating information about packet flows with tags, the information including route information specifying a route through a plurality of interconnected switch nodes;

updating information about the packet flow associated with the computed tag if there is an entry for the computed tag;

creating a new entry in the table if there is no entry for the computed tag; and
removing entries that have not been accessed for a predetermined period from the table.

24. (original) The method of claim 23, wherein the creating further comprises storing data extracted from the packet in the entry.

25. (original) The method of claim 24, wherein the data includes billing information for the packet.

26. (original) The method of claim 24, wherein the packet is sent to a system with resources for analyzing the packet and determining billing information to be associated with the entry for the computed tag.

27. (original) The method of claim 23, wherein the removing further comprises transferring the data associated with a tag to a system with resources for storing information.

28. (original) The method of claim 23, wherein the computing further comprises using a mask of bits of the packet as a seed for a hash code generator.

29. (previously presented) The method of claim 23, wherein the hash code generator is a pseudo random number generator.

30. (previously presented) The method of claim 23, wherein the hash code generator is a shift register with a feedback loop.

31. (previously presented) The method of claim 23, wherein the hash code generator has a nonzero probability of generating the same tag from different input packets.

32. (canceled)

33. (previously presented) The method of claim 23, wherein said fields used to compute said tag are selected from Ethernet and network headers in said packet.

34. (previously presented) The method of claim 23, wherein said fields used to compute said tag are specified by a template, said template specifying fields for a particular protocol.

35. (previously presented) The method of claim 34 wherein said protocol is the real time protocol (RTP).

36. (previously presented) The method of claim 34 wherein said protocol is the hyper-text transfer protocol (HTTP).

37. (previously presented) The method of claim 1 wherein said fields used to compute said tag are specified by a template, said template specifying fields for a particular protocol.

38. (previously presented) The method of claim 37 wherein said protocol is the real time protocol (RTP).

39. (previously presented) The method of claim 37, wherein said protocol is the hyper-text transfer protocol (HTTP).

40. (previously presented) The method of claim 1, wherein each flow detector is loaded with a template for detecting a different protocol.

41-45. (canceled)